

Proposal for a Bond Graph Based Model of Computation in SystemC-AMS

Torsten Mähne*, Alain Vachoux†

Laboratoire de Systèmes Microélectroniques (LSM)
Ecole Polytechnique Fédérale de Lausanne (EPFL), Switzerland

Abstract

SystemC-AMS currently offers modelling formalisms with specialised solvers mainly focussing on the electrical domain. There is a need to improve its modelling capabilities concerning conservative continuous time systems involving the interaction of several physical domains and their interaction with nonconservative digital control components. Bond graphs unify the description of multi-domain systems by modelling the energy flow between the electrical and non-electrical components. They integrate well with block diagrams describing the signal processing part of a system. It is proposed to develop an extension to the current SystemC-AMS prototype, which shall implement the bond graph methodology as a new Model of Computation (MoC).

1 Introduction

Advances in processing technologies allow to pack an increasing number of features in single or multi-chip components (called System-on-a-Chip (SoC) or System-in-a-Package (SiP), respectively). The increasing complexity of the implemented functionalities, the diversity of possible implementations (e.g., digital/analogue/RF hardware, software, Micro-Electro-Mechanical System (MEMS)), and the rapid evolution of the needs (e.g., time to market, product diversity, compliance to standards) ask, on the one hand, for early partitioning decisions that meet design constraints and, on the other hand, for easy reuse and retargeting. Design complexity is usually tackled with the help of modelling and abstraction concepts, which in turn depend on the application domain and are grouped under the generic term of Model of Computation (MoC).

SystemC [8] is a set of C++ classes and methods that provides means to develop behavioural and structural models of digital hardware systems from abstract specifications to Register Transfer Level (RTL) models. SystemC is currently based on the event-driven MoC, which can be used to model abstract behaviours such as transactions between processing units as well as logic behaviours in digital hardware. However, by its very nature, SystemC can be extended to support other MoCs [13] such as Synchronous Data Flow (SDF) or Finite State Machine (FSM).

When it comes to supporting Analogue and Mixed-Signal (AMS) systems, several attempts to extend SystemC have been done and are reviewed in Section 2. This paper presents a proposal to support bond graph based models in SystemC-AMS as a way to model physical systems such as MEMS while providing the necessary abstraction level and simulation speed for system-level design. Section 3 introduces the bond graph methodology and its most recent theoretical advances. Section 4 discusses the application of this methodology to SystemC-AMS, namely the requirements to support bond graphs in SystemC-AMS.

2 AMS extensions for SystemC

Vachoux et al. [15] details the current *SystemC-AMS* prototype, which has been chosen to be developed further under the auspices of the Open SystemC Initiative (OSCI) Analog/Mixed-signal Working Group (AMSWG). Two formalisms, or MoCs, are currently implemented: a variant of the Synchronous Data Flow (SDF) MoC is used to model signal processing dominated behaviours as well as more general continuous-time behaviours using oversampled models and the Linear Network MoC provides a library of linear electrical primitives for describing linear macro models. First experiments for an extension of SystemC-AMS to model conservative elements with non-linear dynamic algebraic equations are presented in Einwich et al. [6]. All these MoCs are synchronised with the discrete-event SystemC simulation kernel through a synchronisation layer, thereby allowing mixed-signal, or mixed-MoC, simulation. The applicability of the current prototype for the modelling of MEMSs, which involve several physical domains under the rule of energy conservation laws, was examined on micromechanical inertial sensors [10]. The mechanical part was chosen to be modelled using a non-conservative block diagram with feedback and simulated using the SDF MoC instead of mapping the mechanical elements like dampers, springs, and masses on their electrical counterparts resistor, capacitor, and inductor. The ΔC - U -converter was modelled using the linear electrical elements.

In Al-Junaid et al. [1] an extended version of SystemC, called *SystemC-A*, is presented. SystemC-A provides constructs to support user-defined Ordinary Differential Equations (ODEs) and Differential Algebraic Equations (DAEs). Conservative systems are analysed

*torsten.maehne@epfl.ch

†alain.vachoux@epfl.ch

using a Modified Nodal Analysis (MNA) by specifying the contributions of each conservative terminal to the Jacobian and the right hand side of the DAE system. However, the presented solution has the drawback that it required modifications to the standard SystemC kernel itself to couple the analogue solver with the discrete-event solver, instead of providing an abstraction layer on top of SystemC to allow the parallel integration of various continuous time MoCs.

Another effort, called *SystemC-WMS*, is presented in Orcioni et al. [12]. It only uses the standard SystemC hierarchical channels to implement analogue modules, which communicate by exchanging energy waves through wave channel interfaces. These interfaces avoid the interconnexion problems commonly found in signal-flow representations of conservative blocks, where the input or output direction of the to one port associated across (e.g., voltage) and through quantities (e.g., current) has to be decided at the implementation time of the module. This is too early because the direction of the information flow is determined from the interconnexion of the modules, which is only known to the simulator at elaboration time. The wave channel methodology avoids this problem since incident waves have always the input role and reflected waves the output role. Parallel and series connexion of modules are achieved through special channels, which dispatch the waves to the modules they connect together. This is similar to the scattering junction of Wave Digital Filters (WDFs). The response of a module to the incident waves is described through a, b parameters used in the WDF theory. The methodology can be extended to circuits with mildly non-linear elements. The observed simulation performance of SystemC-WMS was however limited because of the scheduling of several discrete events for each time step preventing the independent execution of the discrete and continuous parts.

Like in the “classical” AMS-Hardware Description Languages (HDLs) (e.g., VHDL-AMS and Verilog-AMS), SystemC-AMS and SystemC-A use a generalised network model to represent conservative systems, in which modified Kirchoff’s voltage and current laws account for the energy conservation. The generalised networks represent accurately the physical structure of the conservative system but they have the drawback that they don’t visualise the computational structure of the system answering the question: Which quantities act as input and which as outputs? The relation between the across and through quantities at the terminals of each component are described using ODEs or DAEs. The equation setup is quite low-level in SystemC-AMS (overloaded `equation` method, which calculates procedurally the new quantity values from the ones of the previous time step) and SystemC-A (overloaded method, which setups for each time step the Jacobian matrix and right hand side of the DAE to be solved) compared to the direct entry of DAEs in VHDL-AMS. The algorithms used to solve the equation system for the whole generalised network and thus the simulation performance are similar for SystemC-AMS, SystemC-A, and classic AMS-HDLs simulators. One shared problem is that the

solvability of the implemented model is usually only checked insufficiently at model elaboration time: e.g., VHDL-AMS only checks that the number of defined across and through equations matches, which is a necessary but not sufficient condition. In Haase [7] the problem of the possible definition of syntactically correct simulation problems without a solution is discussed from a mathematical point of view to give reasons and establish rules that may help to avoid these difficulties.

3 The Bond Graph Methodology

One way to unify the description of a multi-domain system is the bond graph methodology [4, 9], which is used in mechanical engineering, mechatronics, control theory, and to some extent in power electronics [3]. Each conservative system can be transformed from its domain specific representation (e.g., electrical circuit, mechanical multi-body system, rigid bodies, fluidic networks, thermal networks) to a bond graph representing graphically the energy flow through the multi-domain system. As an example the transformation of a simple electrical circuit to an equivalent bond graph annotated with causality and then derived computational structure is shown in Figure 1.

The bonds (half-arrows) represent the energy flow between the ports of the elements of the graph, each one having associated *effort* e and *flow* f variables, which product gives the power:

$$P(t) = e(t) \cdot f(t) \quad (1)$$

They are also called *power variables* for this reason. By definition, the half-arrow points into the direction, in which the power flows for positive e and f . For the description of dynamic systems, two other variable types are important, which belong to the class of *energy variables*: The (*generalised*) *momentum* $p(t)$ is defined as the time integral of an effort:

$$p(t) = \int_{t_0}^t e(t) dt = p_0 + \int_{t_0}^t e(t) dt \quad (2)$$

The (*generalised*) *displacement* $q(t)$ is defined as the time integral of a flow:

$$q(t) = \int_{t_0}^t f(t) dt = q_0 + \int_{t_0}^t f(t) dt \quad (3)$$

with p_0 and q_0 the initial values of p and q , respectively, at the time t_0 .

The energy $E(t)$ is defined as the time integral of the power $P(t)$:

$$E(t) = \int_{t_0}^t P(t) dt = \int_{t_0}^t e(t)f(t) dt \quad (4)$$

The energy can be also expressed as a function of the energy variables by inserting the differential form of (2) and (3) in (4):

$$E(q) = \int^q e(q) dq \quad E(p) = \int^p f(p) dp \quad (5)$$

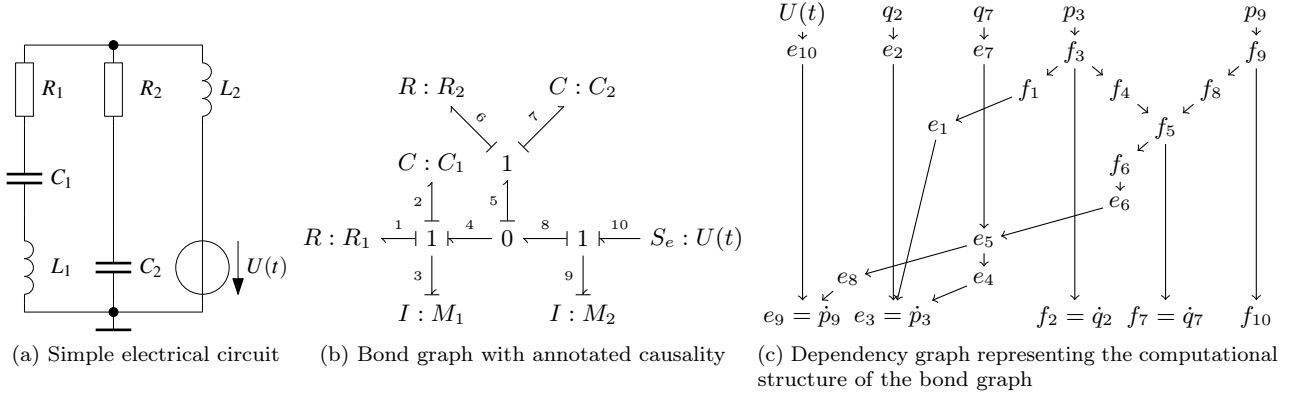


Figure 1: Example of the transformation of an electrical circuit to an equivalent bond graph annotated with causality and then derived computational structure

The definition for the *generalised* power and energy variables is independent of a particular physical domain.

Three generalised 1-port elements represent the resistive R , inertial I , and capacitive C elements independent of the considered physical domain. Energy sources are modelled as effort source S_e and flow source S_f elements. Quantity transformations (also across domain boundaries) are represented through the transformer TF and gyrator GY 2-port elements, while still ensuring energy conservation. Multi-port junction elements represent explicitly the connexion of elements, which are exposed to a common effort (0-junction) or a common flow (1-junction). All elements (Table 1) can have non-linear characteristic equations.

One main advantage of bond graphs is that they can be annotated in a systematic way with the *causality* for each bond (flow-out or effort-out causality), which visualises the computational structure of the bond graph and allows to sort the element equations in the right order for model execution (Figure 1c). The perpendicular stroke at one end of a bond states that at this side the effort variable e is known (it acts as an input) and f can be calculated as a function $f = \Phi_i(e)$. Consequently, the flow f is known on the other side of the bond and acts as an input to a function to calculate the effort e : $e = \Phi_j(f)$. Table 1 summarises the basic elements of a bond graph with the permitted causality assignments and their defining equations. The assigned causalities allow some further formal checks on the model: the number of states and non-states in the system, the presence of algebraic loops during model execution, or if it is an ill-posed model.

The causality assignment allows also for a well integration of bond graphs with signal flow graphs. An example is given in Figure 2 in form of a car wheel model of an electronically controlled suspension system incorporating a semi-active damper and a fast load-leveler [9]. Both ways of representing the systems are given: the “classic” domain-specific way and the way using bond graphs for the energy conservation part and block diagrams for the signal processing part. The blocks in the signal flow graph can take the calculated

power or energy variables as input and can modulate the sources or element parameters of the bond graph.

There are extensions to classical bond graphs, which allow, e.g., hierarchical models with more abstract *word bond graphs* [4] or handling of discrete switching due to external signals by extending the set of bond graph elements with idealised controlled junctions leading to so-called hybrid bond graphs [2, 11]. In Karnopp et al. [9] the generalisation of the basic bond graph elements R , I , and C to multiport fields is introduced and junction structures as an assembly of power-conserving 0- and 1-junctions, transformers TF , and gyrators GY are presented, which allow the effective modelling of complex multiport systems combining structural details and clarity with a convenient visualisation. The usefulness of these concepts is demonstrated on transducer, nonlinear mechanical systems, distributed parameter systems (e.g., a beam), magnetic circuits and thermofluid systems examples. These properties of bond graphs make them attractive for the design and verification of Systems-on-Chips (SoCs) since they unify and thus ease the description of electrical and non-electrical system components and their interactions. They also offer a higher level of abstraction and thus possible run-time advantages over the mentioned classical AMS-HDLs.

Besides specialised bond graph tools such as MTT, 20-sim, and Enport (reviews can be found, e.g., on <http://www.bondgraphs.com/>), there are also efforts to integrate bond graph support in widely used mathematical tools such as MATLAB/Simulink (e.g., BondLab) or Mathematica (e.g., Bond graph tool box for Mathematica). Another approach is to use the capabilities of an AMS-HDL to represent bond graphs. In Cellier and McBride [5] the implementation of causal/a-causal bond graphs of the publicly available BondLib for Modelica/Dymola is presented. Its usage is demonstrated on a position control system involving a hydraulic motor. However, Modelica’s capabilities are weak on the discrete event side, which is one reason why it is not used in the microelectronics community and thus not suitable for mixed-signal SoC design. Pêcheux et al. [14] shows that bond graphs can be represented in VHDL-AMS.

Table 1: Basic elements of a bond graph with their legal causality assignments

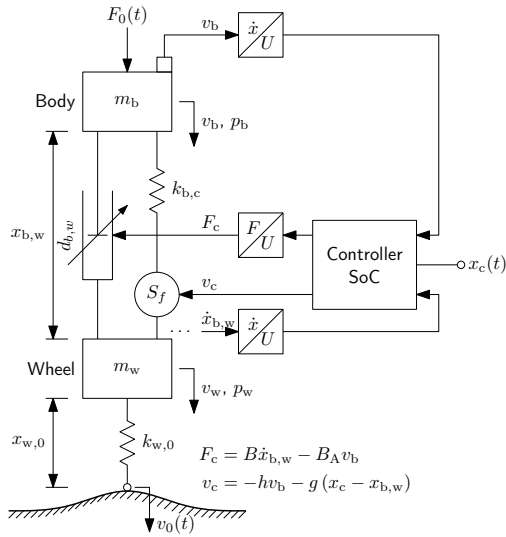
Class	Name	Symbol	Defining relation		Examples from electrical, mechanical, and hydraulic domains
			General case	Linear case	
1-port	effort source	$S_e \xrightarrow[e]{e}$	$e(t)$ given, $f(t)$ arbitrary		voltage, force, pressure sources
	flow source	$S_f \xrightarrow[f]{e}$	$f(t)$ given, $e(t)$ arbitrary		current, velocity, volume flow rate sources
	(generalised) resistor	$\xrightarrow[f]{e} R$ $\xrightarrow[f]{e} R$	$e = \Phi_R(f)$ $f = \Phi_R^{-1}(e)$	$e = Rf$ $f = \frac{1}{R}e$	electrical resistor, translational damper, rotational damper, hydraulic throttle
	(generalised) capacitor	$\xrightarrow[f=\dot{q}]{e} C$ $\xrightarrow[f=\dot{q}]{e} C$	$q = \Phi_C(e)$ $e = \Phi_C^{-1}(q)$	$q = Ce$ $e = \frac{1}{C}q$	electrical capacitor, spring, torsion bars, gravity tank, accumulator
	(generalised) inductor	$\xrightarrow[f]{e=\dot{p}} I$ $\xrightarrow[f]{e=\dot{p}} I$	$p = \Phi_I(f)$ $f = \Phi_I^{-1}(p)$	$p = If$ $f = \frac{1}{I}p$	electric inductor, mass, rotating disk with moment of inertia, section of a fluid-filled pipe with fluid inertia
2-port	(generalised) transformer	$\xrightarrow[f_1]{e_1} TF \xrightarrow[f_2]{e_2}$ $\xrightarrow[f_1]{e_1} TF \xrightarrow[f_2]{e_2}$	$e_1 = me_2, f_2 = mf_1$ $e_2 = \frac{1}{m}e_1, f_1 = \frac{1}{m}f_2$		electrical transformer, ideal rigid lever, gear pair, hydraulic ram
	(generalised) gyrator	$\xrightarrow[f_1]{e_1} GY \xrightarrow[f_2]{e_2}$ $\xrightarrow[f_1]{e_1} GY \xrightarrow[f_2]{e_2}$	$e_1 = rf_2, e_2 = rf_1$ $f_2 = \frac{1}{r}e_1, f_1 = \frac{1}{r}e_2$		electrical gyrator, gyroscope, voice coil transducer
n -port	flow junction, 0-junction, common effort junction	$\begin{array}{c} e_n \downarrow f_n \\ \xrightarrow[f_1]{e_1} 0 \xrightarrow[f_j]{e_j} \\ e_i \uparrow f_i \end{array}$	$e_1 = \dots = e_i = \dots = e_n$ $f_i = -\left(\sum_{j=1, j \neq i}^n f_j\right)$		parallel connexion of electrical conductors; situation involving a single force and n velocities summing up to zero; parallel connexion of hydraulic passages
	effort junction, 1-junction, common flow junction	$\begin{array}{c} e_n \downarrow f_n \\ \xrightarrow[f_1]{e_1} 1 \xrightarrow[f_j]{e_j} \\ e_i \uparrow f_i \end{array}$	$f_1 = \dots = f_i = \dots = f_n$ $e_i = -\left(\sum_{j=1, j \neq i}^n e_j\right)$		series connexion of electrical conductors; dynamic equilibrium of n forces associated with a single velocity; series connexion of hydraulic passages

They are used in the models of a Pb/Fe battery and a complex airbag SoC including a MEMS accelerometer, a digital control, a thermal network, a laser diode, and the chemical reaction to inflate the cushion. The implementation of bond graphs is not very efficient because the support of VHDL-AMS for continuous systems is limited to non-conservative signal flow description based on free quantities and energy conserving generalised networks. Thus, from a bond graph point of view, *effort* and *flow* variables of a port have to be referred from the *across* and *through* quantities defined between two terminals. Satisfying Kirchhoff's voltage and current laws inside the generalised networks leads, in the case of bond graphs where these laws are already ensured through the 0- and 1-junctions, to duplicated equations. Causality cannot be assigned to the bonds and thus not

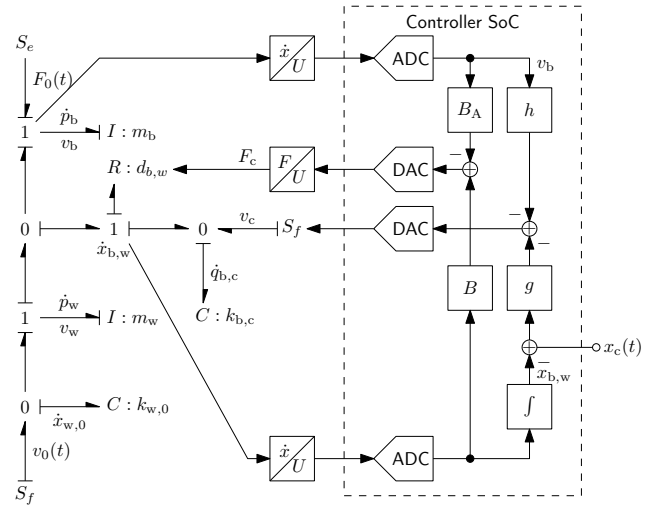
be benefited from. All three aspects have a negative impact on the simulation performance of the model.

4 Application of the Bond Graph Methodology to SystemC-AMS

The goal of this work is to improve the modelling and simulation capabilities of SystemC-AMS regarding conservative continuous time components and their interaction with discrete time (digital) control components. For a high simulation performance, the models should avoid the setup of global DAE systems, which require a complex implicit solver. But this is the case for the current available MoCs for linear and nonlinear elements in energy conserving generalised networks. For non-



(a) "Classic" domain-specific system representation



(b) Representation using bond graphs for the energy conservation part and block diagrams for the signal processing part

Figure 2: Car wheel model of an electronically controlled suspension system incorporating a semi-active damper and a fast load-leveller [adapted from 9]

electrical elements, also the link to the original physical domain gets lost due to the mapping to their electrical counterparts. The models should rather order the equations to allow for a fast procedural execution (similar to the static scheduling of the SDF MoC), which calculates at each step the unknowns for the next ordered equations and thus solves the equation system explicitly. Bond graphs meet these requirements if the assigned causalities are exploited. They keep the link to the modelled physical domain through the units attached to the power/energy variables and the element parameters.

The implementation of the bond graph MoC will profit from the fact that SystemC-AMS is a library on top of the fully-featured C++ language. The causality of a bond graph can be completed in a systematic way at elaboration time using, e.g., the Sequential Causality Assignment Procedure (SCAP) [9] taking into account the known fixed, preferred, or free causality of the element ports. The user can modify the initial causality assignments to guide the algorithm and thus to optimise its result. The derived computational structure allows to sort in advance the element equations in the right order for model execution. It allows also for some formal checks to audit the model at elaboration time: the number of states and non-states in the system, the presence of algebraic loops during model execution, or if it is an ill-posed model. This can guide the designer during model development by providing him more insight into the physical system and drawing thus his attention to potential problems in his design. Algebraic loops can be broken by inserting a one time step delay similarly to what is done in the SDF MoC.

For a clear specification of the physical domain, each variable needs to be annotated with its measurement unit allowing for dimensional analysis and thus discovering of illegal connexion of incompatible ports and calculations involving incompatible quantities. A promising candidate to fulfil this requirement is the quantitative

units library, which allows for unit analysis already at compile time and thus limits its impact on the execution performance and which was recently accepted to be part of the boost project (<http://www.boost.org>).

The integration and synchronisation with the other MoCs, especially the SDF MoC and the Discrete-Event (DE) MoC, which are used to model the signal processing/controlling parts of the system, is needed (Figure 3). This might require small adjustments in the synchronisation layer of the SystemC-AMS prototype. It shall not require modifications of the SystemC kernel. The strong interaction between the analogue and digital parts of a SoC requires to take into account discrete switching of the energy flows inside the bond graphs due to external signals and thus causality changes during the model execution. The research in this field of *hybrid bond graphs*, which incorporate this local switching capability, is still on-going and needs more efforts to find ways to efficiently reassign causality and to regenerate the computational model at runtime when junction switching occurs [2]. Another important aspect is to allow for hierarchical modelling thus implementing aspects of *word bond graphs* [4]. It has to be also evaluated in which way multi-port field elements and junction structures can be supported efficiently by the bond graph extension without sacrificing too much simulation performance.

5 Conclusions and Outlook

This paper proposes the development of a bond graph extension to SystemC-AMS, to improve its modelling capabilities in the field of energy conserving continuous time systems. To this end, the current state of the competing AMS extensions for SystemC was presented with a focus on the SystemC-AMS prototype. It showed that the current solutions for conservative systems modelling

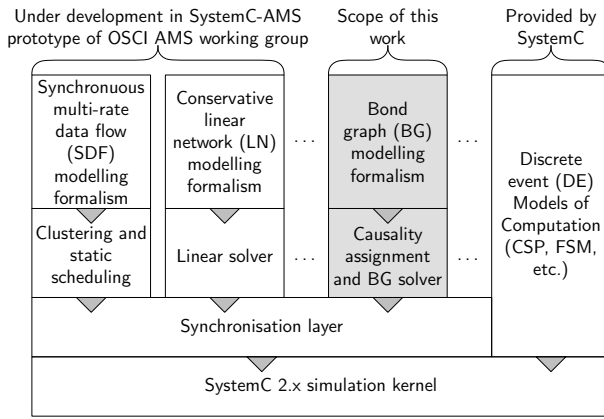


Figure 3: Architecture of SystemC-AMS showing the MoCs provided by the SystemC kernel and the AMS extension as well as the synchronisation layer

are quite low-level with equation setups and analysis methods similar to classic analogue circuits simulators like SPICE, which causes a simulation performance penalty. The bond graph methodology offers advantages for the design and verification of AMS SoCs since it unifies the description of electrical and non-electrical system components on various levels of abstraction. The causality analysis of a bond graph allows to optimise the model execution and gives the designer valuable insight into the computational structure of his model. The requirements for the planned bond graph extension were sketched out and need to be further detailed to derive a specification to be used in the following design and implementation phases.

Acknowledgements

This work has been funded by the Hasler Stiftung under project number 2161.

References

- [1] Hessa Al-Junaid, Tom Kazmierski, and Leran Wang. SystemC-A modeling of an automotive seating vibration isolation system. In *Forum on Specification and Design Languages (FDL) 2006*, pages 107–112, Darmstadt University, Germany, September 19–22 2006. ECSI. ISBN 978-3-00-019710-9.
- [2] Christopher D. Beers, Eric-Jan Manders, Gautam Biswas, and Pieter J. Mosterman. Building efficient simulations from hybrid bond graph models. In *Proceedings of the 2nd IFAC Conference on Analysis and Design of Hybrid Systems*, pages 71–76, Alghero, Italy, June 7–9 2006. URL <http://www.diee.unica.it/~adhs06/CD/PAPERS/WB1.3.pdf>.
- [3] K. Besbes. Modelling semiconductor devices using bond graph techniques. In *Proceedings of the IEEE International Symposium on Industrial Electronics (ISIE) 1997*, volume 2, pages 201–205, Guimaraes, July 7–11 1997. doi: 10.1109/ISIE.1997.648934.
- [4] Peter Breedveld. Bond graphs. In *Proceedings of the Geoplex Summer School 2003*, pages 1–27, Bertinoro, Italy, July 2003. URL http://www-lar.deis.unibo.it/euron-geoplex-sumsch/files/lectures_1/Breedveld/Breedveld_03_BGConcepts.pdf.
- [5] François E. Cellier and Robert T. McBride. Object-oriented modeling of complex physical systems using the Dymola bond-graph library. In *Proceedings of the 6th SCS International Conference on Bond Graph Modeling and Simulation (ICBGM) 2003*, pages 157–162, Orlando, Florida, January 19–23 2003. SCS. URL http://www.inf.ethz.ch/personal/fcellier/Pubs/BG/icbgm_03_bglib.pdf.
- [6] Karsten Einwich, Jens Bastian, Christoph Clauss, Uwe Eichler, and Peter Schneider. SystemC-AMS extension library for modeling conservative nonlinear dynamic systems. In *Forum on Specification and Design Languages (FDL) 2006*, pages 113–120, Darmstadt University, Germany, September 19–22 2006. ECSI. ISBN 978-3-00-019710-9.
- [7] Joachim Haase. Rules for analog and mixed-signal VHDL-AMS modeling. In *Proceedings of the 6th International Forum on Specification & Design Languages (FDL) 2003*, pages 98–107, University of Frankfurt, Germany, September 23–26 2003. ECSI. URL <http://www.eas.iis.fhg.de/publications/papers/2003/023/paper.pdf>.
- [8] IEEE Standard 1666-2005. *IEEE Standard 1666-2005, SystemC Language Reference Manual*. IEEE, March 2006. URL <http://standards.ieee.org/reading/ieee/std/dasc/1666-2005.pdf>.
- [9] Dean C. Karnopp, Donald L. Margolis, and Ronald C. Rosenberg. *System Dynamics: Modeling and Simulation of Mechatronic Systems*. Wiley, 4 edition, January 2006.
- [10] Erik Markert, Hendrik Zeun, Göran Herrmann, Dietmar Müller, and Ulrich Heinkel. SystemC-AMS-Modell eines DeltaC-U-Wandlers für ein Inertialnavigationssystem. In *Tagungsband der 9. ITG/GMM-Fachtagung Analog'06 „Entwicklung von Anlogschaltungen mit CAE-Methoden“*, pages 243–248, Dresden, 27.–29. September 2006. ISBN 3-8007-2988-1. URL ftp://kreuzas.infotech.tu-chemnitz.de/pub/doc2006/Dresden_analog06.pdf.
- [11] Pieter J. Mosterman. HyBrSim – a modeling and simulation environment for hybrid bond graphs. *Journal of Systems and Control Engineering*, 216(1):35–46, February 1 2002. URL <http://elib.dlr.de/11830/01/hybrsm-mosterman.pdf>.
- [12] S. Orcioni, G. Biagetti, and M. Conti. SystemC-WMS: A wave mixed-signal simulator. In *Proceedings of the 8th International Forum on Specification & Design Languages (FDL) 2005*, pages 61–72, EPFL, Lausanne, Switzerland, September 27–30 2005. ECSI.
- [13] Hiren Patel and Sandeep K. Shukla. *SystemC Kernel Extensions for Heterogeneous System Modeling*. Kluwer Academic Publishers, The Netherlands, 2004. ISBN 978-1402080876.
- [14] François Pêcheux, Bruno Allard, Christophe Lallement, Alain Vachoux, and Hervé Morel. Modeling and simulation of multi-discipline systems using bond graphs and VHDL-AMS. In *Proceedings of the International Conference on Bond Graph Modeling and Simulation (ICBGM) 2005*, pages 149–155, New Orleans, USA, January 23–27 2005. The Society for Modeling and Simulation International (SCS). URL <http://infoscience.epfl.ch/getfile.py?mode=best&recid=54126>.
- [15] A. Vachoux, C. Grimm, and K. Einwich. Extending SystemC to support mixed discrete-continuous system modeling and simulation. In *Proceedings of the IEEE International Symposium on Circuits and Systems (ISCAS) 2005*, pages 5166–5169, May 23–26 2005. doi: 10.1109/ISCAS.2005.1465798.